



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/944,311	08/30/2001	Peter D. J. Dennis	SUN-P6268-PIP	2172
57960	7590	06/06/2006	EXAMINER	
SUN MICROSYSTEMS INC. C/O PARK, VAUGHAN & FLEMING LLP 2820 FIFTH STREET DAVIS, CA 95616			FOWLKES, ANDRE R	
			ART UNIT	PAPER NUMBER
			2192	

DATE MAILED: 06/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.		Applicant(s)	
	09/944,311		DENNIS ET AL.	
	Examiner		Art Unit	
	Andre R. Fowlkes		2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 February 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,2,4,7-10,12,15-18,20,23 and 24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,2,4,7-10,12,15-18,20,23 and 24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 2/16/06 has been entered.
2. Claims 1,2,4,7-10,12,15-18,20,23 and 24 are pending. Claims 1, 2, 9, 10, 17 and 18 have been amended.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
4. The rejection of claims 9 & 17 under 35 U.S.C. 112, second paragraph, is withdrawn, in view of applicant's amendment.
5. Claim 1 recites the limitation "examining the plurality of structure definitions in the storage structure" in lines 16-17. Claim 1 does not disclose that "structure definitions" are stored in the storage structure; therefore, there is insufficient antecedent basis for this limitation in the claim. It appears to be a typo. The examiner is interpreting the limitation "examining the plurality of structure definitions in the storage structure " as --

examining the plurality of *data* structure definitions in the storage structure --, to further prosecution.

6. Accordingly, claims 2,4,7 and 8 are rejected as being dependent on a rejected base claim.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1, 2, 4, 7 & 8 rejected under 35 U.S.C. 103(a) as being unpatentable over applicants admitted prior art, (AAPA), in the background section of the instant application in view of Parlante, "Linked List Basics", Stanford CS Education Library. The PGPUB application paragraph and line numbers are used to cite the AAPA reference.

As per claim 1, AAPA discloses a **method to facilitate debugging computer code within an operating system kernel** (§ 0006:1-4, "In an effort to provide debugging capabilities for the operating system kernel, engineers have created a

modular debugger, which can facilitate debugging the operating system kernel”),
comprising:

- receiving an operating system source file containing a data structure definition (§ 0007:3-5, “examines the (operating system) source files of the operating system kernel to determine the data structures (and definitions) within the kernel”),
wherein the operating system source file contains a plurality of data structures, and wherein the operating system source file is part of an operating system kernel (§ 0006:5-10, to gather data for the data structures within the operating system (source file, and) ... display or print the gathered data”).

- searching the operating system source file for the data structure definition (§ 0007:3-5, “examines the source files of the operating system kernel to determine the data structures within the kernel”),

- upon finding the data structure definition, saving the data structure definition in a storage structure (§ 0006:7-8, “this gathered data can then be saved in the computer system’s memory”),

- automatically generating a new operating system source code to display a data structure through execution of a source generator program, wherein the new operating system source code is created using the data structure definition (§ 0006:5-10, “(generating new) ... source code, which is custom designed, (per the data structure definition), to gather data for the data structures within the operating system (and) ... display or print the gathered data”, and merely using a computer to automate a known process does not by itself impart nonobviousness to the invention. See *Dann v.*

Johnston, 425 U.S. 219, 227-30, 189 USPQ 257, 261 (1976); *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958)),

- **wherein automatically generating the new operating system source code includes generating source code to traverse data structures** (§ 0006:5-10, “(generating new) ... source code, which is custom designed, to gather data for the data structures within the operating system”, and merely using a computer to automate a known process does not by itself impart nonobviousness to the invention. See *Dann v. Johnston*, 425 U.S. 219, 227-30, 189 USPQ 257, 261 (1976); *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958)),

- **and wherein generating the new operating system source code involves: examining the plurality of data structure definitions in the storage structure to locate a cross-reference between data structures; and generating the new operating system source code for the plurality of data structures** (§ 0007:3-5, “examines the source files of the operating system kernel to determine the data structures within the kernel”, and § 0006:5-10, “(generating new) ... source code, which is custom designed, (per the data structures and their cross reference information), to gather data for the data structures within the operating system”).

- **compiling the new operating system source code into an executable module; installing the executable module into a modular debugger** (§ 0008:1-5, “after creating this source code, the operator compiles the source code into an executable module , which is then inserted into the modular debugger”),

- during execution of the modular debugger, displaying a content of the data structure to a user of the modular debugger using the executable module, whereby the user is able to view the content of the data structure (§ 0008:3-5, "(the modular debugger is operable) to gather data from the data structures within the kernel while the kernel is executing", and § 0008:7-10, "This gathered data can then be ... display(ed)").

AAPA doesn't explicitly disclose generating source code to **walk a linked list** of data structures.

However, Parlante, in an analogous environment, discloses generating source code to **walk a linked list** of data structures (p. 17:6-7, "A very frequent technique in linked list (source) code is to iterate (i.e. walk) a pointer over all the nodes (i.e. data structures) in a (linked) list").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Parlante into the system of AAPA to generating source code to **walk a linked list** of data structures. The modification would have been obvious because one of ordinary skill in the art would have wanted to generate source code that is able to completely gather data from any type of data structure used (in this case the data structure used is a linked list).

As per claim 2, the rejection of claim 1 is incorporated and further, AAPA discloses that **receiving the operating system source file includes receiving a**

plurality of operating system source files (§ 0007:3-5, "examines the source files of the operating system kernel to determine the data structures within the kernel").

As per claim 4, the rejection of claim 1 is incorporated and further, AAPA discloses that **saving the data structure definition in the storage structure includes saving the plurality of data structures in the storage structure** (§ 0007:3-5, "examines the source files of the operating system kernel to determine the data structures within the kernel", and § 0006:7-8, "this gathered data (structures) can then be saved in the computer system's memory").

As per claim 7, the rejection of claim 6 is incorporated and further, AAPA discloses that **displaying the content of the data structure includes displaying the content of the linked list of data structures** (§ 0006:5-10, "(generating new) ... source code, which is custom designed, (per the data structures and reference data), to gather data for the data structures (i.e. linked list of data structures) within the operating system (and) ... display or print the gathered data").

As per claim 8, the rejection of claim 1 is incorporated and further, AAPA discloses that **the data structure definition includes one of a tree, a linked list, a doubly linked list, and a queue** (§ 0006:6, "data structures (i.e. trees, linked lists, doubly linked lists, queues").

Art Unit: 2192

9. Claims 9, 10, 12, 15-18, 20, 23 & 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over applicants admitted prior art, (AAPA), in the background section of the instant application, in view of Parlante, "Linked List Basics", Stanford CS Education Library, further in view of Vazquez et al., (Vazquez), U.S. Patent No. 6,763,515.

As per claim 9, AAPA discloses **a method to facilitate debugging computer code within an operating system kernel** (§ 0006:1-4, "In an effort to provide debugging capabilities for the operating system kernel, engineers have created a modular debugger, which can facilitate debugging the operating system kernel"), **comprising:**

- **receiving an operating system source file containing a data structure definition** (§ 0007:3-5, "examines the source files of the operating system kernel to determine the data structures within the kernel"), **wherein the operating system source file contains a plurality of data structures, and wherein the operating system source file is part of an operating system kernel** (§ 0006:5-10, to gather data for the data structures within the operating system (source file, and) ... display or print the gathered data").

- **searching the operating system source file for the data structure definition** (§ 0007:3-5, "examines the source files of the operating system kernel to determine the data structures within the kernel"),

- upon finding the data structure definition, saving the data structure definition in a storage structure (§ 0006:7-8, “this gathered data can then be saved in the computer system’s memory”),

- automatically generating a new operating system source code to display a data structure through execution of a source generator program, wherein the new operating system source code is created using the data structure definition (§ 0006:5-10, “(generating new) ... source code, which is custom designed, (per the data structure definition), to gather data for the data structures within the operating system (and) ... display or print the gathered data”),

- wherein automatically generating the new operating system source code includes automatically generating source code to traverse data structures (§ 0006:5-10, “(generating new) ... source code, which is custom designed, to gather data for the data structures within the operating system”),

- and wherein generating the new operating system source code involves: examining the plurality of data structure definitions in the storage structure to locate a cross-reference between data structures; and generating the new operating system source code for the plurality of data structures (§ 0007:3-5, “examines the source files of the operating system kernel to determine the data structures within the kernel”, and § 0006:5-10, “(generating new) ... source code, which is custom designed, (per the data structures and their cross reference information), to gather data for the data structures within the operating system”).

- **compiling the new operating system source code into an executable module; installing the executable module into a modular debugger** (§ 0008:1-5, “after creating this source code, the operator compiles the source code into an executable module , which is then inserted into the modular debugger”),

- **during execution of the modular debugger, displaying a content of the data structure to a user of the modular debugger using the executable module, whereby the user is able to view the content of the data structure** (§ 0008:3-5, “(the modular debugger is operable) to gather data from the data structures within the kernel while the kernel is executing”, and § 0008:7-10, “This gathered data can then be ... display(ed)”).

AAPA doesn't explicitly disclose generating source code to **walk a linked list** of data structures.

However, Parlante, in an analogous environment, discloses generating source code to **walk a linked list** of data structures (p. 17:6-7, “A very frequent technique in linked list (source) code is to iterate (i.e. walk) a pointer over all the nodes (i.e. data structures) in a (linked) list”).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Parlante into the system of AAPA to generating source code to **walk a linked list** of data structures. The modification would have been obvious because one of ordinary skill in the art would

have wanted to generate source code that is able to completely gather data from any type of data structure used (in this case the data structure used is a linked list).

AAPA doesn't explicitly disclose **a computer readable storage medium storing instructions that when executed by a computer cause the computer to perform** the algorithm listed above.

However, Vazquez, in an analogous environment, discloses **a computer readable storage medium storing instructions that when executed by a computer cause the computer to perform** an algorithm (col. 4:7-8, "providing a system and method for automatically generating a program to perform an ... algorithm").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Vazquez into the AAPA/Parlante system to have **a computer readable storage medium storing instructions that when executed by a computer cause the computer to perform an** algorithm. The modification would have been obvious because one of ordinary skill in the art would want the use the well known technique of automating a manual algorithm using computer software, to attain speed and consistency.

As per claims 10, 12, 15 & 16, this is a computer readable medium version of the claimed method discussed above, in claims 2, 4, 7 & 8, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA (§ 0006:1-0008:5).

As per claims 17, 18, 20, 23 & 24, this is an apparatus version of the claimed method discussed above, in claims 9, 10, 12, 15 & 16, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see AAPA (¶ 0006:1-0008:5).

Response to Arguments

10. Applicants arguments have been considered but they are not persuasive.

In the remarks, the applicant has argued substantially that:

1) Neither, AAPA, Parlante nor Vazquez, alone or in combination disclose automatically locating data structures within operating system source code and generating new operating system source code to assist in debugging the operating system, at p. 8:18-9:3.

Examiner's response:

1) The examiner disagrees with applicant's characterization of the applied art. The AAPA/Parlante/Vazquez combination discloses automatically locating data structures within operating system source code, (AAPA, ¶ 0007:3-5, "examines the (operating system) source files (i.e. code) of the operating system kernel to determine the data structures within the kernel"), and generating new operating system source code to assist in debugging the operating system, (AAPA, ¶ 0006:5-10, "(generating

new) ... source code, which is custom designed, to gather data for the data structures within the operating system".) Moreover, merely using a computer to automate a known process does not by itself impart nonobviousness to the invention. See *In re Venner*, 262 F.2d 91, 95, 120 USPQ 193, 194 (CCPA 1958). See also *Dann v. Johnston*, 425 U.S. 219, 227-30, 189 USPQ 257, 261 (1976).


Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ARF



TUAN DAM
SUPERVISORY PATENT EXAMINER